ES 50

Saagar Deshpande & Randy Miller

April 12, 2012

1 Overview

We would like to make a car powered by Arduino that is remotely controlled by a Microsoft Kinect.

Initially, our goal will be to have 1 of us get a skeleton Kinect application up and running that can respond to the movements that we want, and the have the other create an Arduino car that can respond to some input, possibly hard-coded into the application beforehand. Then we will work on interfacing the application and the Arduino via a USB connection. Then, we will fine tune our application and our Arduino code/circuit for performance. The only materials we will need are an Arduino kit, a BOE BOT, a Kinect, and Visual Studio.

Ultimately, we expect to have an Arduino car respond (hopefully correctly) to Kinect input such that it can drive around like a normal remote-controlled car.

2 Features/Deliverables

2.1 Deliverables

- Robot car can move in all reasonable directions
- We can send instructions to the car from an Arduino
- We have a Kinect application that can register basic gestures/voice commands
- We can send input to the Aruduino from the Kinect application

2.2 Extras

- Wireless connectivity (via IR)
- Sophisticated Kinect Application
- Tuning/Optimization
- remote camera relayed to computer
- go to specified location (AI)/give instructions

3 Design

3.1 Materials

- Kinect
- Kinect SDK
- Visual Studio

- Arduino Kit and Software, including circuit materials
- BOE BOT

3.2 Pipeline

1. Kinect processes input (voice/gesture) and translates to some pre-defined protocol
   (a) Kinect application registers input
   (b) Kinect application translates input to some pre-defined protocol
   (c) Kinect application sends data to Aruduino over USB connection
2. The Arduino receives and parses messages
   (a) Arduino receives message in C
   (b) The data is quickly parsed into robot car instructions
   (c) The resultant data is passed to the robot car
3. The parsed data is then interpreted and executed by the car motors

4 Timeline

4.1 Stage 1: Design and Basic Implementations

1. Design and create circuit for Arduino and BOE BOT
2. Create Arduino code to control the car with simple hard-coded instructions
3. Design and create basic bare-bones Kinect application that can register (not recognize) input
4. Design interface between Kinect and Arduino
5. Design interface (if necessary) between Arduino and the car

4.2 Stage 2: Complicating the implementations and Adding Interfaces

1. Adapt Arduino code to accept input that's formatted to adapt to the interface
2. Test Arduino car on dynamic inputs, maybe via command line on a host computer
3. Adapt Kinect application to recognize and parse input (not just register)
4. Create classes that fulfill the Kinect/Arduino interface
5. Add input-processing that translates input to the interface classes

4.3 Stage 3: Connecting the Components and Testing

1. Adapt the Arduino to accept data from the Kinect
2. Connect the Kinect to the Arduino
3. Test

4.4 Stage 4: Testing, Optimization, and Extra Features

1. More testing
2. Add polish and optimize
3. Add extra features, in this order
   (a) Wireless connectivity using IR sensors and receivers
   (b) Make the Kinect application fancy
   (c) Make the Kinect application parse instructions
   (d) All other extras

5 Notes

Our source code will be hosted on code.seas.harvard.edu, meaning that we will be using Git as for our souce code management needs.

We will be using Sandcastle to compile documentation for our Kinect application, and we will put any documentation for the Arduino into a PDF document.